UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/816,213	04/01/2004	Judy M. Gehman	03-1002/L13.12-0246	7306

7590 09/16/2009

LSI Logic Corporation
Leo J. Peters
MS D-106
1621 Barber Lane
Milpitas, CA 95035

EXAMINER	
RAMPURIA, SATISH	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
09/16/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/816,213
Filing Date: April 01, 2004
Appellant(s): GEHMAN ET AL.

David D. Brush
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 06/25/2009 appealing from the Final Office action mailed 10/08/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

Based on the information supplied by the Appellants, and to the best of Appellants' legal representative's knowledge, the real party in the interest is the assignee, LSI Corporation of Milpitas, California.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct and listed below for convenience.

Claims pending: 1-20

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2002/0100029	BOWEN	7-2002
6,425,116	DUBOC ET AL.	7-2002
6,829,754	YU ET AL.	12-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 5-11, and 13-20 are rejected under 35 U.S.C. 103(a) as being uneatable over US Publication No. 2002/0100029 to Bowen (hereinafter, Bowen) in view of US Patent No. 6,425,116 to Duboc et al. (hereinafter, Duboc).

Per claim 1:

Bowen discloses:

A method for coding a hardware description of a peripheral device, the method comprising:

configuring a function block to instantiate multiple instanced of the peripheral device within a single chip design (paragraph [0011] "the function in the FPGA is shared amongst all its uses…the configuration of the FPGA is duplicated for each use"; paragraph [0041] "a number of hardware and software resources from a single behavioral description of the system" and paragraph [0040] "…a hardware/software codesign system…codesign system receives a behavioral description of the target electronic system and automatically partitions the required functionality between hardware and software…hardware and the processor for the software can be formed on an FPGA")

the hardware description of the peripheral device having options associated with different configurations of the peripheral device (paragraph [0009] "The hardware configuration information is utilized to configure a Field Programmable Gate Array (FPGA) for compiling the function to the FPGA… invention could also be applied to compile functions to reconfigurable logic devices other than FPGAa (i.e., those device having different configurations)" and (paragraph [0010] "generates hardware configuration information from the processor instructions… to configure an FPGA such that the function is compiled to the FPGA"); and

wherein the options are selected without modification to the hardware description (paragraph [0036] “a hardware compiler for producing from those parts of the specification partitioned to hardware a register transfer level description for configuring configurable logic resources”; paragraph [0011] "the function in the FPGA is shared amongst all its uses…the configuration of the FPGA is duplicated for each use").

compiling the hardware description to produce a structural model comprising each instance of the peripheral device with the selected options for that instance (paragraph [0138] “compilation stages of the process flow are software or hardware… module 212… allocates any behavioral parts of the hardware description, and at module 216 compiles the software description to assembly code… also writes a parameterized description… designed by the user”).

Bowen does not explicitly disclose selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another.

However, Duboc discloses in an analogous computer system selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another (col. 8, lines 33-39 “the integrated circuit design via selection of a compile option from the GUI window, resulting in the execution of a script engine 152 in the HDL Integrator tool that processes a check script 154 developed by a developer, and used to verify the parameters input by a user” and col. 10, lines 31-34 “Memory Integrator tool from Philips Semiconductor, that generates models for a memory compiler that

generates customized memory components suitable for interfacing within a custom DSP integrated circuit").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another as taught by Duboc into a method and computer program product are provided for compiling a C function to a reconfigurable logic device as taught by Bowen. The modification would be obvious because of one of ordinary skill in the art would be motivated to selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another to provide an apparatus, program product and method for use in automating the design of a custom DSP integrated circuit from a preexisting DSP core block and one or more additional circuit blocks interfaced with the DSP core block as suggested by Duboc (col. 2-3, lines 65-67 and 1-4).

Per claim 2:

The rejection of claim 1 is incorporated and further, Bowen discloses:

The method of claim 1 wherein the step of selecting comprises:

passing a parameter value to the function block at compile time for each instantiation of the hardware peripheral (paragraph [0109] "RTL descriptions are passed straight through to the RTL synthesizer e.g. a Handel-C compiler."); and

instance the peripheral device using code according to the parameter value (paragraph [0111] "Behavioral descriptions will be scheduled in such a way that the block of code will execute within that number of cycles, when possible. An error is generated if it is not possible").

Per claim 3:

The rejection of claim 1 is incorporated and further, Bowen discloses:

The method of claim 1 wherein the configuration options comprises at least one of peripheral design functions, peripheral design pin widths, or peripheral design interface pin outs (paragraph [0037] "The system can include a width adjuster for setting and using a desired data word size, and this can be done at several points in the desired process as necessary").

Per claim 5:

The rejection of claim 1 is incorporated and further, Bowen discloses:

The method of claim 1 wherein the step of configuring comprises:

configuring the function block with local runtime constants adapted to be overridden individually at compile time (paragraph [0134] "hardware and software compilers 304, 306, and may be used or overridden…functions which must be supplied by its subclasses… compile method on the hardware compiler class compiles the description to hardware by converting the input description to an RTL description; the compile method on the Processor A compiler compiles a description to machine code which can run on Processor A").

Per claim 6:

The rejection of claim 5 is incorporated and further, Bowen discloses:

The method of claim 5 wherein the step of selecting comprises

overriding selected runtime constants at compile time to select between the variable options for each instance of the peripheral device (paragraph [0134] "hardware and software compilers 304, 306, and may be used or overridden…functions which must be supplied by its subclasses… compile method on the hardware compiler class compiles the description to hardware by converting the input description to an RTL description; the compile method on the Processor A compiler compiles a description to machine code which can run on Processor A").

Per claim 7:

Bowen discloses:

A method for coding a reusable hardware description of a peripheral device, the method comprising:

configuring a function block to instantiate multiple instances of the peripheral device within an integrated circuit design (paragraph [0041] "a number of hardware and software resources from a single behavioral description of the system"), the reusable hardware description of the peripheral device having options selectable at compile time (paragraph [0011] "the configuration of the FPGA is duplicated for each use, so that the function is used as an inline function" and paragraph [0009] "hardware configuration information is utilized to configure a Field Programmable Gate Array (FPGA) for compiling the function to the FPGA" and paragraph [0241] "OOP components are reusable software modules which present an interface that conforms to an object model and which are accessed at run-time through a component integration architecture"); and

instantiating the multiple instances of the peripheral device on the integrated circuit design by programmatically (paragraph [0031] "The codesign system comprising means for receiving a specification of the functionality, partitioning means for partitioning implementation of the functionality between (a) and (b) and for customizing the hardware and/or the machine in accordance with the selected partitioning of the functionality" and paragraph [0241] "OOP components are reusable software modules which present an interface that conforms to an object model and which are accessed at run-time through a component integration architecture"); compiling the reusable hardware description to produce a structural model comprising the multiple instance of the peripheral device, each with the selected options and resulting configuration for that instance [0138] "compilation stages of the process flow are software or hardware… module 212… allocates any behavioral parts of the hardware description, and at module 216 compiles the software description to assembly code… also writes a parameterized description… designed by the user").

Bowen does not explicitly disclose selecting between the options at compile time for each instance of the peripheral device so that at least two of the instances have different configurations.

However, Duboc discloses in an analogous computer system selecting between the options at compile time for each instance of the peripheral device so that at least two of the instances have different configurations (col. 8, lines 33-39 "Once a user has provided input to the GUI window, the user initiates generation of the integrated circuit design via selection of a compile option from the GUI window, resulting in the execution of a script engine 152 in the

HDL Integrator tool that processes a check script 154 developed by a developer, and used to verify the parameters input by a user" and col. 10, lines 31-34 "Memory Integrator tool from Philips Semiconductor, that generates models for a memory compiler that generates customized memory components suitable for interfacing within a custom DSP integrated circuit")

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of selecting between the options at compile time for each instance of the peripheral device so that at least two of the instances have different configurations as taught by Duboc into a method and computer program product are provided for compiling a C function to a reconfigurable logic device as taught by Bowen. The modification would be obvious because of one of ordinary skill in the art would be motivated to selecting between the options at compile time for each instance of the peripheral device so that at least two of the instances have different configurations to provide an apparatus, program product and method for use in automating the design of a custom DSP integrated circuit from a preexisting DSP core block and one or more additional circuit blocks interfaced with the DSP core block as suggested by Duboc (col. 2-3, lines 65-67 and 1-4).

Per claim 8:

The rejection of claim 7 is incorporated and further, Bowen discloses:

The method of claim 7 wherein the variable options are selected without modification to the reusable hardware description (paragraph [0036] "a hardware compiler for producing from those parts of the specification partitioned to hardware a register transfer level description for configuring configurable logic resources").

Per claim 9:

The rejection of claim 7 is incorporated and further, Bowen discloses:

The method of claim 7 wherein the step of configuring comprises:

adding one or more peripheral devices based on desired features of the reusable hardware to the integrated circuit design at compile time (Bowen discloses without modification to the configurations the option are selected for each use. Further, Bowen teaches a system of using the single code block coupled with a hardware tailored for a specific peripheral device to simply 'share' the block to allow using the dedicated software with greater flexibility for multiple functionality see paragraph [0241]).

Per claim 10:

The rejection of claim 7 is incorporated and further, Bowen discloses:

The method of claim 7 wherein the step of configuring comprises: instantiating peripheral devices onto the integrated circuit according to the reusable hardware description wherein the configuration of each instance is unique based on a design parameter (paragraph [0111] "Behavioral descriptions will be scheduled in such a way that the block of code will execute within that number of cycles, when possible. An error is generated if it is not possible" and paragraph [0036] "a hardware compiler for producing from those parts of the specification partitioned to hardware a register transfer level description for configuring configurable logic resources").

Per claim 11:

The rejection of claim 10 is incorporated and further, Bowen discloses:

The method of claim 10 wherein the design parameter comprises a signal width of the peripheral device (paragraph [0037] "The system can include a width adjuster for setting and using a desired data word size, and this can be done at several points in the desired process as necessary").

Per claim 13:

The rejection of claim 7 is incorporated and further, Bowen discloses:

The method of claim 7 wherein the step of configuring further comprises:

configuring the function block with parameters local in scope, the parameters adapted to be overridden individually at compile time (paragraph [0134] "hardware and software compilers 304, 306, and may be used or overridden…functions which must be supplied by its subclasses… compile method on the hardware compiler class compiles the description to hardware by converting the input description to an RTL description; the compile method on the Processor A compiler compiles a description to machine code which can run on Processor A").

Per claim 14:

The rejection of claim 13 is incorporated and further, Bowen discloses:

The method of claim 13 wherein the step of selecting comprises overriding selected runtime constants at compile time to select between the options for each instance of the peripheral device

(paragraph [0134] "hardware and software compilers 304, 306, and may be used or overridden…functions which must be supplied by its subclasses… compile method on the hardware compiler class compiles the description to hardware by converting the input description to an RTL description; the compile method on the Processor A compiler compiles a description to machine code which can run on Processor A").

Per claim 15:

The rejection of claim 7 is incorporated and further, Bowen discloses:

The method of claim 7 wherein the step of configuring comprises:

passing a parameter value to the function block at compile time for each instance of the peripheral device (paragraph [0109] "RTL descriptions are passed straight through to the RTL synthesizer e.g. a Handel-C compiler."); and

instantiating the peripheral device using the reusable hardware description according to the parameter value (paragraph [0111] "Behavioral descriptions will be scheduled in such a way that the block of code will execute within that number of cycles, when possible. An error is generated if it is not possible").

Claims 16-20 are the method claim corresponding to method claims 1, 2, 5, 6, and 11 respectively, and rejected under the same rational set forth in connection with the rejection of claims 1, 2, 5, 6, and 11respectively, above, as noted above.

Claims 4 and 12 rejected under 35 U.S.C. 103(a) as being unpatentable over Bowen, Duboc in view of US Patent No. 6,829,754 to Yu et al. (hereinafter, Yu).

Per claim 4:

The rejection of claim 1 is incorporated and further, neither Bowen nor Duboc explicitly discloses tying strap pins to power or ground.

However, Yu discloses in an analogous computer system tying strap pins to power or ground (col. 11, lines 2-5 "Straps do not have a minimum width, defined as the width of the power pin the strap is connecting to. If the strap is smaller than the power pin it connects, a warning will be issued").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of tying strap pins to power or ground as taught by Yu into the method of using a computer program to reconfigure the logic devices as taught by the combination system of Bowen and Duboc. The modification would be obvious because of one of ordinary skill in the art would be motivated to strap the power or ground pins so that the power related problems can be avoid (col. 2, lines 8-11).

Per claim 12:

The rejection of claim 7 is incorporated and further, neither Bowen nor Duboc explicitly discloses defining further the function block by tying strap pins to ground or to power.

However, Yu discloses in an analogous computer system defining further the function block by tying strap pins to ground or to power (col. 11, lines 2-5 "Straps do not have a

minimum width, defined as the width of the power pin the strap is connecting to. If the strap is smaller than the power pin it connects, a warning will be issued").

The feature of defining further the function block by tying strap pins to ground or to power would be obvious for the reasons set forth in the rejection of claim 4.

(10) Response to Argument

Appellant argued that:

Bowen Does Not Disclose Different Configurations Instantiated

in a Single IC Chip Design

Examiner respectfully disagrees. Bowen discloses the function in the FPGA is shared amongst all its uses. The configuration is duplicated for each use, so that the function is used an inline function (paragraph [0011]). Bowen discloses a system of using the single code block coupled with a hardware tailored for a specific peripheral device to simply 'share' the block to allow using the dedicated software with greater flexibility for multiple functionality (paragraph [0241]). Thus, allowing the FPGA to have different configurations for greater flexibility and multiple functionalities. Further, appellant indicated that Examiner mistakenly refers to paragraphs [0040-0041] however, these paragraphs support that from a single behavioral description a partition across a number of hardware and software resources performed (i.e., the number of hardware having different configuration) for providing the system of great flexibility, see also paragraphs [0053-0054]. This method of coding overcomes the similar shortcoming as described in the background section of Applicant's prior art that the software block has to be

reconfigured repeatedly (paragraph [0005]) to use with a separate hardware device as explained in the Applicant's "Background of the Invention" section.

Options are Not Selectable Without Modification To The Hardware Description

Examiner respectfully disagrees. Bowen discloses in paragraph [0241] that reusable software modules to simply share the code i.e., without modification to code as claimed. Bowen discloses the function in the FPGA is shared amongst all its uses. The configuration is duplicated for each use, so that the function is used an inline function (paragraph [0011]). Thus, Bowen discloses duplicating i.e., without modification to the configurations the option are selected for each use.

Bowen Does Not Disclose Selecting Between the Options at Compile Time for Each Instantiation of the Peripheral Device

Examiner respectfully disagrees. Appellant does not provide any reasoning for their arguments and simply stated that office action acknowledges that Bowen does not disclose the above claimed limitations. As explained in the office action that the above limitations are taught by the combination of Bowen and Duboc. More particularly, Duboc discloses selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another, see col. 8, lines 33-39 which discloses the integrated circuit design via *selection of a compile option* from the GUI window, resulting in the execution of a script engine 152 in the HDL Integrator tool that processes a check script 154

developed by a developer, and used to verify the parameters input by a user and col. 10, lines 31-34 which discloses Memory Integrator tool from Philips Semiconductor, that generates models for a memory compiler that generates customized memory components suitable for interfacing within a custom DSP integrated circuit.

Duboc et al. Fails to Disclose That Two Different Instantiations Can Have Two Different Configurations Selectable at Compile Time

Examiner respectfully disagrees. The above limitations are taught by the combination of Bowen and Duboc. More particularly, Duboc discloses that designing a custom DSP integrated circuit using a DSP builder template that operates as a front end to a design reuse tool. The template permits both the selection of one or more optional circuit blocks and the customization of one or more customizable circuit blocks to be performed via a user interface. Moreover, once any optional circuit blocks are selected and any customization parameters are selected for a given customizable circuit block, a custom DSP integrated circuit is automatically built by customizing any such customizable block and interfacing a preexisting DSP core block with any selection optional circuit blocks and customized circuit blocks, see col. 5, line 55 to col. 6, line 24.

Duboc specifically discloses on Fig.5 that selecting different configuration at compile time. Duboc's Fig.5 discloses the usage of a **DSP builder template** to automate the **development of a custom DSP integrated circuit** consistent with the invention. The sequence of operations illustrated in FIG. 5 at 140 may be utilized to **generate custom DSP integrated circuits using a DSP Builder template** in the manner discussed herein. The DSP Builder

template is represented in the HDL Integrator environment by a collection of GUI components, check scripts and build scripts.

The overall process begins as illustrated in block 142 by constructing one or more graphical user interface windows via the design tool window builder. Input to the design tool window builder is a series of GUI components 144 developed by a developer and utilized to received user input for interacting with the template. The output of builder 142 is one or more GUI windows 146, which are configured to receive user input as illustrated at 148 as a user interacts with the window to define one or more parameters for use in generating a custom DSP integrated circuit consistent with the invention. One suitable template window is illustrated at 150. Once a user has provided input to the GUI window, **the user initiates generation of the integrated circuit design via selection of a compile option from the GUI window**, resulting in the execution of a script engine 152 in the HDL Integrator tool that processes a check script 154 developed by a developer, and used to verify the parameters input by a user. In particular, the check script essentially is processed by the script engine to analyze the user options and verify that such options are consistent with the available options for the DSP subsystem. Based upon the output of the script engine, it is determined whether the inputs provided by the user are acceptable. If any errors are detected, an error window is displayed at 158, and control returns to block 148 to receive modified input from the user to correct any errors in the inputs previously provided thereby. If the inputs are acceptable, control passes to block 160 to execute the script engine once again to process a build script 162 that processes the parameters to output one or more output files 164 representing the output of the template. **Utilizing the HDL Integrator tool described above, the GUI components file 144 is in the format of a ".TPL" or ".STPL"**

file as processed by the design reuse tool to generate a graphical interface window and any required subwindows. Check script file 154 is in the format of an ".HPC" file that contains the code to check all of the user inputs and generate any error or other warning messages. The build script file 162 is in the **format of an ".HTF" file functioning as the generic scripts** that **generate the VHDL, VERILOG or other high definition language files with the parameters that have been specified by the user in the template window**. It will be appreciated, however, that in other environments, alternate source files may be utilized to control a design reuse tool in the manner described herein, see col. 8, lines 12-65 also col. 10, lines 25-45.

The modification would be obvious because of one of ordinary skill in the art would be motivated to select between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another to provide use in automating the design of a custom DSP integrated circuit from a preexisting DSP core block and one or more additional circuit blocks interfaced with the DSP core block as suggested by Duboc (col. 2-3, lines 65-67 and 1-4).

With respect to appellant arguments for dependent claims 2 and 3 (similarly claims 15, 17, and 20), Appellant indicated that Applicants does not understand why examiner cited Bowen paragraph [0111] for limitations "passing a parameter value to the function block at compile time for each instance of the hardware peripheral". Apparently, Appellants looking at the wrong paragraph because for the above limitations paragraph [0109] was cited **NOT** paragraph [0111] of Bowen, see the rejection above and the prosecution history.

With respect to appellant arguments for claim 3, Appellant indicated that in Bowen the width adjustment applied is the same every instance and cited paragraph [0114]. However, Bowen discloses the width adjuster as a variable which can be changed as needed, see paragraph [0037] and further, Variables are declared with explicit bit widths and the operators working on the variables work with an arbitrary precision. This allows efficient implementation in hardware. For instance a statement which declares the width of variables (in this case the program counter pc, the instruction register ir, and the top of stack tos) is as follows: unsigned 12 pc, ir, tos. The width of the data path of the processor in the target system may be declared, or else is calculated by the partitioner 208 as the width of the widest variable which it uses, see paragraphs [0062-0064].

With respect to appellant arguments for claims 7 and 16, the arguments are similar to those presented with respect to claim 1 and thus given the same rational as set forth with related to claim 1.

With respect to appellant arguments for claims 4 and 12 that the combination of Bowen, Duboc and Yu does not teach or make obvious the inventions recited in claims 4 and 12 for the similar reasons as discussed above with respect to the independent claims. Examiner respectfully disagrees, the arguments are similar to those presented with respect to claim 1 and thus given the same rational as set forth with related to claim 1. Further, as acknowledged by the office action that neither Bowen nor Duboc explicitly discloses tying strap pins to power or ground. However, Yu discloses the limitations tying strap pins to power or ground, see col. 11, lines 2-5.

The examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the rejection points out that the motivation to tying strap pins to power or ground would be to avoid the power related problems during the design of hardware description level.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Satish Rampuria

/Satish Rampuria/

Examiner, Art Unit 2191

Conferees:

Wei Y. Zhen

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191

Lewis A. Bullock, Jr.

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193